(12) UK Patent Application (19) GB (11) 2 372 172 (13) A

(43) Date of A Publication 14.08.2002

(21) Application No 0113214.1

(22) Date of Filing 31.05.2001

(51) INT CL[7]
H04L 12/56 // H04L 12/24 29/06

(52) UK CL (Edition T)
H4K KTKX

(56) Documents Cited
EP 1028600 A2 US 6134239 A
US 6034945 A US 5546389 A

(58) Field of Search
UK CL (Edition S) H4K KTKX
INT CL[7] H04L 12/24 12/56 29/06
Online: WPI, EPODOC, JAPIO

(71) Applicant(s)
Telefonaktiebolaget L M Ericsson (publ)
(Incorporated in Sweden)
SE-126-25, Stockholm, Sweden

(72) Inventor(s)
Mats Sågfors

(74) Agent and/or Address for Service
Marks & Clerk
4220 Nash Court, Oxford Business Park South,
OXFORD, OX4 2RU, United Kingdom

(54) Abstract Title
Congestion handling in a packet data network

(57) A method of controlling the entry of data packets into a buffer present in a packet transmission link. The method comprises defining a first fixed threshold level and a second variable threshold level for the packet queue within the buffer, and for each data packet arriving at the buffer, dropping that packet or a packet already contained in the buffer if the current buffer queue exceeds said first or second threshold level. The second variable threshold level is adjusted depending upon (a) whether or not a packet is dropped, and (b) the relative values of the first and second thresholds and the queue size.

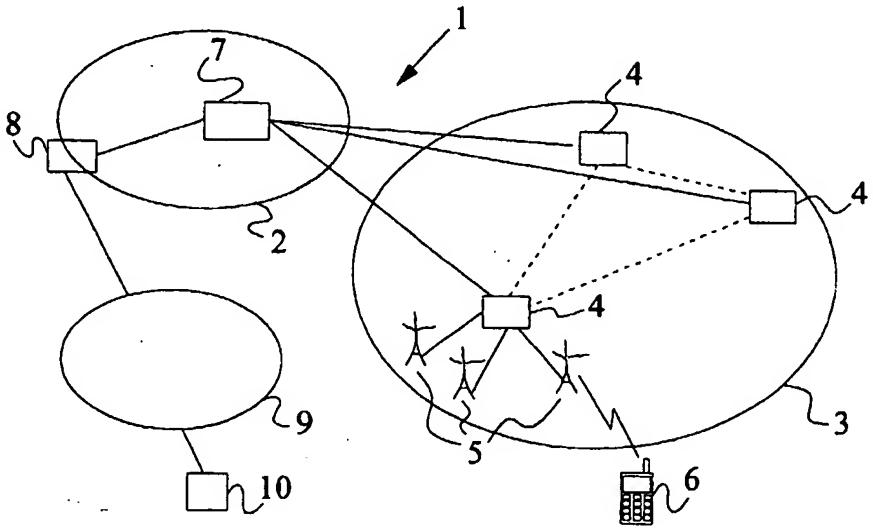

Figure 1

GB 2 372 172 A

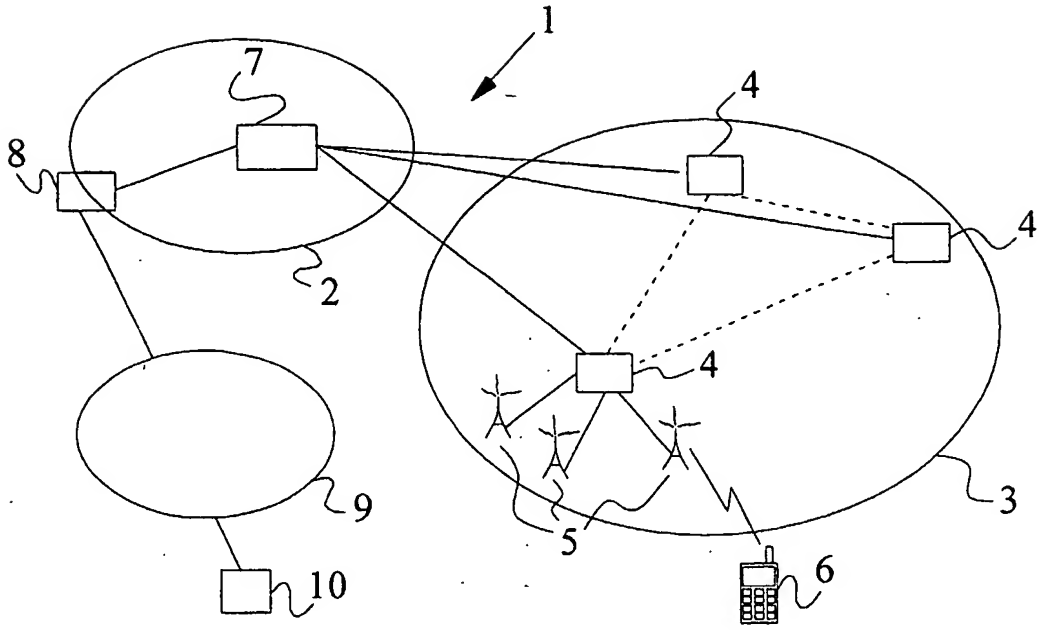

Figure 1

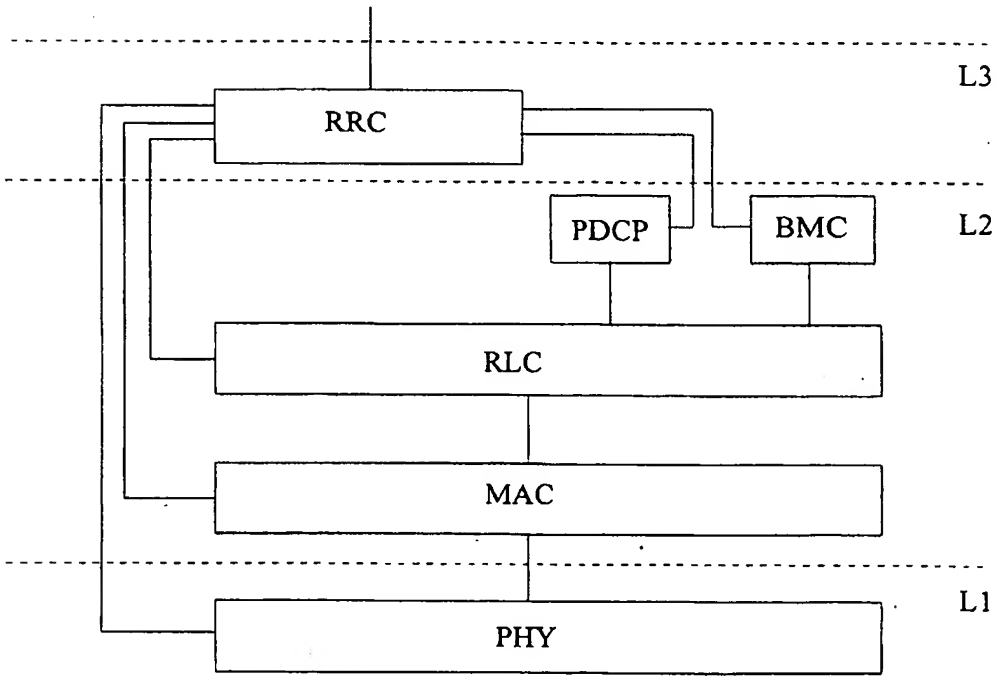

Figure 2

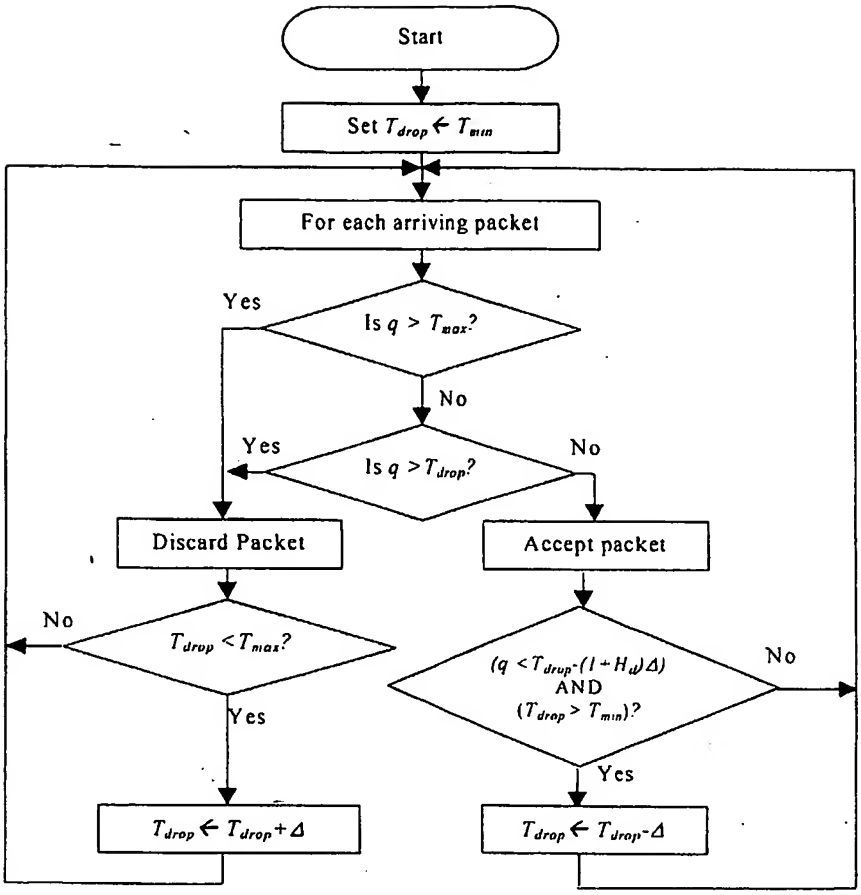
Start
Set $T_{drop} \leftarrow T_{min}$
For each arriving packet
Yes
Is $q > T_{max}$?
No
Yes
Is $q > T_{drop}$?
No
Discard Packet
Accept packet
No
$T_{drop} < T_{max}$?
Yes
$(q < T_{drop}-(1+H_d)\Delta)$ AND $(T_{drop} > T_{min})$?
No
Yes
$T_{drop} \leftarrow T_{drop}+\Delta$
$T_{drop} \leftarrow T_{drop}-\Delta$

Figure 3

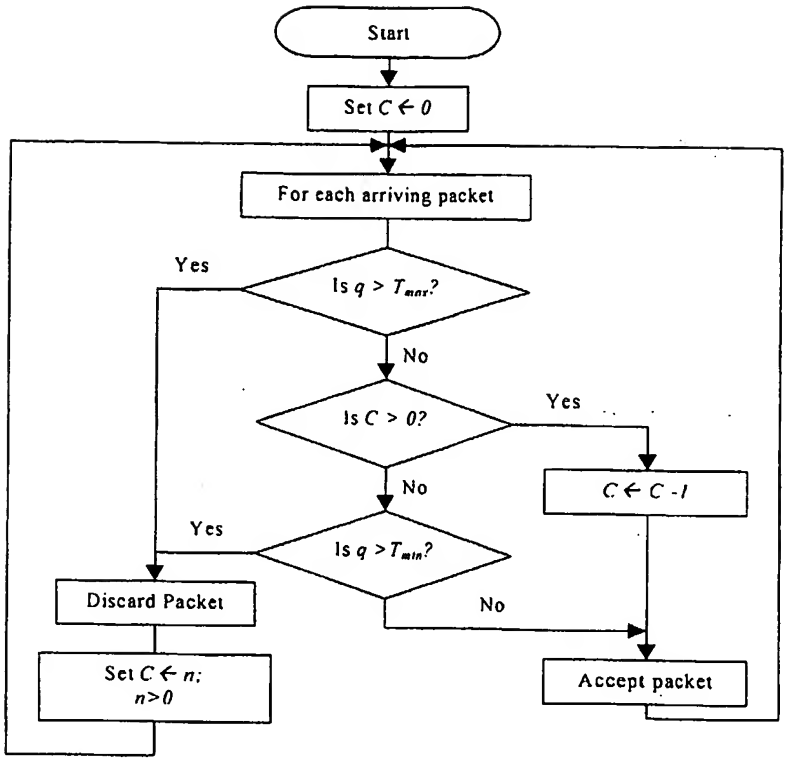
Start
Set $C \leftarrow 0$
For each arriving packet
Yes
Is $q > T_{max}$?
No
Is $C > 0$?
Yes
$C \leftarrow C - 1$
No
Yes
Is $q > T_{min}$?
Discard Packet
No
Set $C \leftarrow n$;
$n>0$
Accept packet

Figure 4

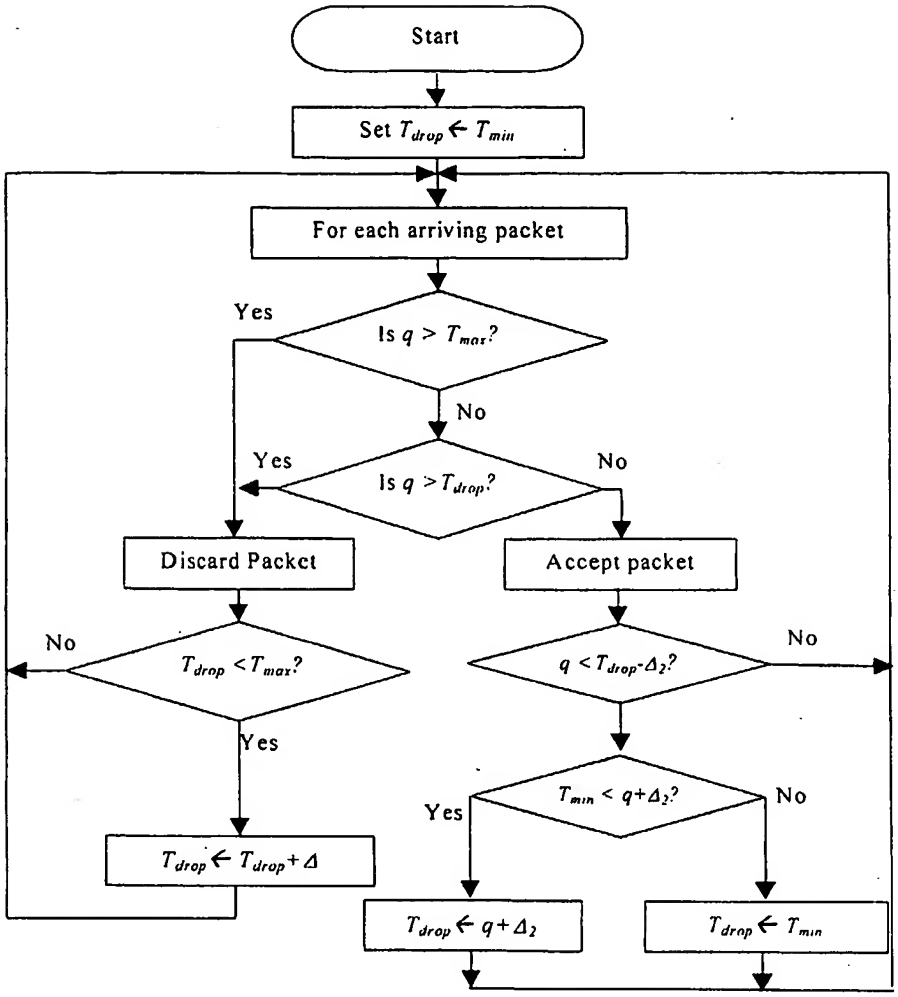
Start
Set $T_{drop} \leftarrow T_{min}$
For each arriving packet
Yes
Is $q > T_{max}$?
No
Yes
Is $q > T_{drop}$?
No
Discard Packet
Accept packet
No
$T_{drop} < T_{max}$?
No
$q < T_{drop}-\Delta_2$?
Yes
Yes
$T_{min} < q+\Delta_2$?
No
$T_{drop} \leftarrow T_{drop}+\Delta$
$T_{drop} \leftarrow q+\Delta_2$
$T_{drop} \leftarrow T_{min}$

Figure 5

2372172

CONGESTION HANDLING IN A PACKET DATA NETWORK

Field of the Invention

The present invention relates to the handling of congestion in a packet data network and more particularly to the early detection of congestion and the implementation of mechanisms for obviating the consequences of congestion.

Background to the Invention

In data packet based communication systems, i.e. in which information to be transmitted is divided into a plurality of packets and the individual packets are sent over a communication network, it is known to provide queue buffers at various points in the network. A buffer may be a sending or input buffer (i.e. a buffer for data packets that are to be sent over a link) or a receiving or output buffer (i.e. a buffer for data packets that have already been sent over a link).

Packets for transporting data may also be called by any of a variety a variety of names, such as protocol data packets, frames, segments, cells, etc., depending on the specific context, the specific protocol used, and certain other conventions. In the context of the present document, all such packets of data shall generically be referred to as data packets. The procedures for placing data packets into a queue, advancing them in the queue, and removing data packets from the queue are referred to as "queue management".

A phenomenon that is known in data packet transmission networks is that of congestion. Congestion implies a state in which it is not possible to readily handle the number of data packets that are required to be transported over that connection or link. As a consequence of congestion at a given link, the number of data packets in a queue buffer associated with said link will increase. In response to a congestion condition, it is known to implement a data packet dropping mechanism referred to as "drop-on-full". According to this mechanism, upon receipt of a new data packet at the queue buffer, a queue length related parameter, such as the actual queue length or the average queue

length, is compared to a predetermined threshold. If the predetermined threshold is exceeded, then a data packet is dropped. The threshold indicates the "full" state of the queue.

The data packet which is dropped can be the newly arrived packet, in which case the mechanism is called "tail-drop". Besides the technique of tail-drop, it is also known to perform a so-called "random-drop", where a data packet already in the queue is selected according to a random function, or a so-called "front-drop", where the first data packet in the queue is dropped. Such drop-on-full mechanisms not only serve to reduce the load on the congested link, but also serve as an implicit congestion notification to the source and/or destination of the data packet.

The so-called "Transmission Control Protocol" (TCP) is a commonly used protocol for controlling the transmission of data packets (or "packets") over an IP network. When a TCP connection between peer hosts is initiated, TCP starts transmitting data packets at a relatively low rate. The transmission rate is slowly increased in order to avoid causing an overflow at routers of the IP network (which would result in the loss of data packets and the need to resend these lost packets). The rate at which data packets can be transmitted is defined by two variables, *cwnd* and *ssthresh*. TCP uses acknowledgement messages to control the transmission rate, and is constantly probing the link for more transmission capacity.

The variable *cwnd* defines the number of unacknowledged data packets which the TCP sender may have in "flight" at any given time. At the beginning of a communication, *cwnd* is set at a low value (e.g. one segment) and the system is in a "slow start" mode. Following receipt of the first acknowledgement from the receiver, *cwnd* is increased in size by one packet (to two packets). Two further packets are then sent. When an acknowledgement is received by the sender for each further packet, *cwnd* is increased by one packet. Once both packets have been acknowledged, the size of *cwnd* is four packets. This process is repeated resulting in an exponential opening of the congestion window. The variable *ssthresh* is initially set to some fixed level (e.g. 65535 bytes), and the slow start mode continues until *cwnd* > *ssthresh*. Thereafter, a "congestion avoidance" mode is entered during which *cwnd* is increased by just 1/*cwnd* each time a

successful transmission acknowledgement is received. The variable *cwnd* has an upper limit defined either by the sender or by an advertisement message sent from the receiver.

If congestion occurs as indicated by a timeout (of a controlling timer at the sender), *ssthresh* is set to one half of the previous value of *cwnd*, and *cwnd* is set to 1. Thus, the slow start mode is re-entered and continued until such time as the transmission rate (defined by *cwnd*) reaches half the rate which last caused congestion to occur. Thereafter, the congestion avoidance mode is entered. If congestion is indicated by receipt of a third duplicate acknowledgements by the sender (indicating that a given data packet has not been received by the receiver despite the receipt of three subsequent segments), *ssthresh* is set to one half of the previous value of *cwnd* whilst shrinks to *ssthresh*. Receipt of three duplicate acknowledgements causes the TCP sender to retransmit the missing data packet using the "fast retransmit" mechanism. After retransmitting the missing data packet, fast recovery takes over. The value of *cwnd* is set to *ssthresh*+3, and is increased by 1 packet for each additional duplicate acknowledgement received. An acknowledgement which acknowledges the retransmitted data packet sets *cwnd* to *ssthresh*, putting the sender back into congestion avoidance mode.

In any IP packet transmission path, bottlenecks will occur which limit the transmission rate of the available transmission route (or link). In conventional networks, bottlenecks may occur for example at IP routers. Routers handle bottlenecks by using buffers to queue incoming data. If the tail dropping mechanism described above is used to deal with congestion, there is a high probability that two or more packets from the same connection will be dropped. The loss of two or more packets from the same sending window of a TCP connection may cause the TCP sender to enter the slow start mode. This timer-triggered loss recovery may lead to under-utilisation of the link, in particular when the link incorporates significant delays. This in turn results in a waste of link resources and perceived poor link performance on the part of the user.

The tail dropping mechanism may also cause problems due to "global synchronisation". This phenomenon arises when several TCP connections simultaneously reduce their

load. The queue serving the connections may be drained resulting in large fluctuations in the buffer level.

In order to avoid the adverse effects of tail dropping, methods to detect congestion before the absolute limit of the queue is reached have been developed. In general these Early Congestion Detection methods make use of one or more queue threshold levels to determine whether or not a packet arriving at a queue should be accepted or dropped. In the so-called "Random Early Detection" method, RED, [IETF RFC2309], a minimum threshold level T_{min} and a maximum threshold level T_{max} are defined. If the queue size remains below the minimum threshold level, all packets arriving at the queue are accepted and placed at the back of the queue. If the queue size exceeds the maximum threshold level, all packets arriving at the queue are dropped. If the queue size is between the maximum and minimum thresholds, packets are dropped with a certain probability. However, this tends to result in only a fraction of the large set of TCP connections (that share the congested router) reducing their load simultaneously. For a queue fill level greater than the maximum threshold, RED works according to the conventional tail drop scheme. The key to the RED algorithm lies in the early congestion notifications that are transmitted to randomly chosen TCP users by dropping a few packets probabilistically when the queue level exceeds the minimum threshold. Since the congestion feedback is transmitted to a limited number of link users, global synchronisation can be avoided.

In order to allow for a certain level of short-term fluctuations in the queue caused by packet bursts (a property inherent to IP transmissions), the RED algorithm does not operate on the instantaneous queue level, but rather on a moving average measure of the queue level $q_{avg}(.)$. When using the RED algorithm, there are four parameters that have to be set by the operator; a queue filter constant w_q, the two queue thresholds T_{min} and T_{max}, and the parameter p_{max} which defines the maximum probability for a packet discard when $T_{min}<q_{avg}(.)<T_{max}$.

RED is reported to work well with high capacity routers. A large number of TCP connections are required to overload such capacity. RED relies heavily on this fact: at congestion there are a large number of connections sharing the queue. It thus makes

sense to "signal" congestion to only a small fraction of users at the same time in order to avoid global synchronisation.

In the paper "Random Early Detection Gateways for Congestion Avoidance" by Sally Floyd and Van Jacobson, IEEE/ACM Transactions on networking, August 1993, an extensive discussion of the RED algorithm is given, where the minimum threshold min_{th}, maximum threshold max_{th}, and the maximum probability max_p are all set as fixed parameters. Regarding the choice of min_{th} and max_{th}, it is mentioned that the optimum values for these thresholds depend on the desired average queue size, and the optimal value for max_{th} depends in part on the maximum average delay over the link. Furthermore, it is stated that max_{th} should at least be twice as large as min_{th}.

In an internet document discussing the setting of RED parameters, published by Sally Floyd at http://www.acir.org/floyd/REDparameter.txt, it is mentioned that the optimum value for fixing min_{th} will depend partly on the link speed, the propagation delay and the maximum buffer size.

In the article "Techniques for eliminating packet loss in congested TCP-IP networks" by Wu-chang Feng et al., November 1997, a so-called adaptive RED is proposed, in which the probability parameter max_p is adapted to the traffic load. Although the detailed algorithm described in this document uses fixed thresholds, it is indicated that the threshold values could also be made dependent on the input traffic. A similar proposal is made in the article "A self configuring RED gateway" by Wu-chang Feng et al., Infocom '99, March 1999.

Another proposal for improving RED is made in WO 00/60817, in which a differentiation is introduced between traffic originating from rate adaptive applications that respond to packet loss. This document suggests introducing at least two drop precedent levels, referred to as "in profile" and "out profile". Each drop precedent level has its own minimum threshold min_{th} and/or maximum threshold max_{th}.

From WO 00/57599 a queue management mechanism is known in which drop functions are selected according to ingress flow rate measurements and flow profiles.

From US-6,134,239 a method of rejecting ATM cells at an overloaded load buffer is known. The concept of RED is mentioned. According to this document, a first threshold related to the overloaded buffer queue, and a second threshold associated with a specific connection are monitored, and incoming packets are dropped for the specific connection if both thresholds are exceeded.

US-5,546,389 describes a method for controlling access to a buffer and is specifically concerned with ATM buffers. The use of one or more thresholds and the dynamic control of such thresholds is mentioned, where the dynamics are determined on the basis of incoming and outgoing traffic.

EP-1 028 600 describes a buffer management scheme with dynamic queue length thresholds for ATM switches. A common threshold is dynamically updated every time a new cell arrives, where the new value is determined based on traffic condition.

Another improvement proposal for RED is described in EP-0 872 988, which has the object of providing isolation when connections using different TCP versions share a bottleneck link. The solution proposed in this document is the use of bandwidth reservation guarantees for each connection. If one connection is being under-utilised, then another connection may use a part of the under-utilised connection's bandwidth. When the connection needs to reclaim its buffer space a predetermined package dropping mechanism is operated, such as a longest queue first (LQF) mechanism.

It will be appreciated that mechanisms such as RED may be employed to trigger the "marking" of data packets when a buffer queue starts to be full. Thus, rather than dropping a packet, the mechanism may add a tag to a packet forwarded to a receiver to notify the receiver that action should be taken to avoid congestion. The receiver may in turn notify the sender. Alternatively, a marked data packet or other notification may be returned directly to the sender.

Whilst much of the state of the art in this area is concerned with IP network routers and the like, the problems of congestion and buffer queue management also arise in mobile communication systems such as cellular telephone networks.

Statement of the Invention

Buffering in mobile communication systems, such as occurs in a UMTS network at the RLC entity of an RNC, must be able to deal satisfactorily with links providing low bandwidth and high latencies. Low bandwidth implies that one or at most a few TCP connections may congest the link. High latency means that TCP will respond slowly when data packets are discarded. The probabilistic view adopted by the RED mechanism is not easily applied to this type of buffering problem. RED applies a low-pass filter on the measured queue level to track long-term trends in the queue size by filtering out high frequency variations due to bursts of data packets. This is not a problem for high capacity routers where large fluctuations relative to the buffer size are not expected. However, for low capacity links, the queue level may build up very quickly relative to the buffer size. For such links (where congestion may develop rapidly) the RED algorithm has two properties that may delay the notification of congestion to the TCP sender:

1) The use of a low pass filter in measuring queue size causes a delay in providing congestion feedback to the TCP sender(s);

2) The probabilistic way of discarding packets, which means that several packets may be accepted into the queue after congestion is detected but before the congestion is notified to the sender.

It has been recognised by the inventors of the present invention that congestion should be notified to the TCP sender(s) as soon as congestion is detected. The congestion notification procedure may be implicit, such as dropping a packet, or explicit by marking a congestion flag in the data packet.

According to a first aspect of the present invention there is provided a method of controlling the entry of data packets into a buffer present in a packet transmission link, the method comprising:

defining a first fixed threshold level and a second variable threshold level for the packet queue size within the buffer; and

for each data packet arriving at the buffer, performing a congestion avoidance procedure if the current buffer queue size exceeds said first or second threshold level, and adjusting said second variable threshold level depending upon (a) whether or not a packet is dropped and (b) upon the relative values of the first and second thresholds and the queue size.

Preferably, said packet transmission link is an IP packet transmission link.

Typically, said congestion avoidance procedure involves dropping the newly arrived data packet, or a data packet already held in the queue. Alternatively however, the procedure may involve adding a congestion marker to the arrived packet before it is added to the queue, or sending some other form of notification to the packet sender and/or receiver.

Assuming that the congestion avoidance procedure involves dropping the newly arrived data packet, or a data packet already held in the queue, if the current buffer queue size is less than both of said first and second threshold levels, or the converse is true but a packet already held in the queue is dropped, the newly arrived packet is added to the queue.

The second variable threshold level is initialised to a predetermined minimum threshold level, which is less than said first fixed threshold level. Preferably, the second variable threshold level is adjusted by incrementing or decrementing the level by a fixed amount. The amount by which the variable threshold is incremented may be the same as the amount by which it is decremented. Alternatively however the amount by which the variable threshold is incremented may be greater than the amount by which it is decremented. In yet another alternative, the variable threshold may be incremented by a fixed amount when that is deemed to be appropriate. When it is deemed appropriate to reduce the variable threshold, the variable threshold is reduced to within some predetermined value in excess of the queue size. In this way, the variable threshold tracks the queue size when the queue size is falling.

Preferably, the second variable threshold level is incremented following receipt of a packet if the packet is discarded and the second variable threshold level does not exceed the first threshold level. If the packet is discarded and the second variable threshold level does exceed the first threshold level, the second variable threshold level is not changed.

Preferably, the second variable threshold level is decremented following receipt of a packet if the packet is accepted, the queue size is less than the second variable threshold level by some predefined amount, and the second variable threshold level is greater than said minimum threshold level. If the packet is accepted, and the queue size exceeds the second variable threshold less said predefined amount or the second variable threshold level is less than said minimum threshold level, the second variable threshold level is not changed.

Preferably, in the event that a decision is made to drop a packet, that packet is a packet from the front, or near the front, of the buffer queue. The newly received packet is entered into the back of the buffer queue. The advantage of this approach is that a packet sender is notified more quickly of congestion in the transmission link.

Preferably, the IP packets belong to a TCP/IP connection, with the packets arriving at the buffer being transmitted there by a TCP sender. The buffer may be located at the sending entity in a layer beneath the TCP layer, at an intermediate point in the connection, or at the receiving entity again in a layer beneath the TCP layer. It will be appreciated that the present invention may also provide benefits for other transmission control protocols, particularly where those protocols implement a congestion control mechanism.

The buffer may be associated with a wireless communication network, for example a mobile telephone network, for the purpose of buffering IP packets prior to their transmission over an air interface. In particular, the network may be UMTS network or equivalent, with the buffer being an RLC buffer. The buffer is located at the sending

entity which may be on the network side, e.g. at a Radio Network Controller, or at a sending User Equipment (mobile station).

According to a second aspect of the invention there is provided apparatus for buffering data packets in a packet transmission link, the apparatus comprising:

a buffer for containing a queue of data packets;

an input for receiving data packets;

a memory storing a first fixed threshold level and a second variable threshold level for the packet queue within the buffer; and

a controller arranged, for each data packet arriving at the buffer, to perform a congestion avoidance procedure if the current buffer queue exceeds said first or second threshold level, and to adjust said second variable threshold level depending upon (a) whether or not a packet is dropped, and (b) the relative values of the first and second thresholds and the queue size.

The apparatus may be arranged for use with a wireless network such as a mobile telephone network. Where the network is a third generation network such as a UMTS network, the apparatus may be a RNC of the radio access network or a user equipment. Alternatively, the apparatus may be for example a router of an IP network or apparatus of an Internet Service Provider (ISP).

According to a third aspect of the present invention there is provided a method of controlling the entry of data packets into a buffer associated with the radio layers of a wireless network, the buffer storing packets prior to their transmission over the radio interface, the method comprising:

defining minimum and maximum threshold levels for the packet queue within the buffer; and

for each data packet arriving at the buffer,

performing a congestion avoidance procedure if the current buffer queue exceeds said maximum threshold level; or

not performing said procedure if the current buffer queue is less than said minimum threshold level; or

if the current buffer queue lies between said maximum and minimum thresholds, performing said procedure only if the arriving packet is the *n*th packet arriving at the buffer since the arrival of the last packet which caused said procedure to be performed.

According to a fourth aspect of the present invention there is provided apparatus for use in a wireless network and comprising:

a buffer for storing data packets

an input for receiving data packets;

a memory storing minimum and maximum threshold levels for the packet queue within the buffer; and

a controller arranged for each data packet arriving at the buffer to,

perform a congestion avoidance procedure if the current buffer queue exceeds said maximum threshold level; or

not perform said procedure if the current buffer queue is less than said minimum threshold level; or

if the current buffer queue lies between said maximum and minimum thresholds, performing said procedure only if the arriving packet is the *n*th packet arriving at the buffer since the arrival of the last packet which caused said procedure to be performed.

Brief Description of the Drawings

Figure 1 illustrates schematically a UMTS network comprising a core network and a UTRAN;

Figure 2 illustrates certain radio protocol layers existing at a RNC of the UTRAN of Figure 1;

Figure 3 is a flow diagram illustrating a process for controlling a queue in an RLC buffer of an RNC node of the UTRAN of Figure 1;

Figure 4 is a flow diagram illustrating an alternative process for controlling a queue in an RLC buffer of an RNC node of the UTRAN of Figure 1; and

Figure 5 is a flow diagram illustrating yet another alternative process for controlling a queue in an RLC buffer of an RNC node of the UTRAN of Figure 1.

Detailed Description of a Preferred Embodiment

Figure 1 illustrates schematically a UMTS network 1 which comprises a core network 2 and a UMTS Terrestrial Radio Access Network (UTRAN) 3. The UTRAN 3 comprises a number of Radio Network Controllers (RNCs) 4, each of which is coupled to a set of neighbouring Base Transceiver Stations (BTSs) 5. BTSs are sometimes referred to as Node Bs. Each Node B 5 is responsible for a given geographical cell and the controlling RNC 4 is responsible for routing user and signalling data between that Node B 5 and the core network 2. All of the RNCs are coupled to one another. A general outline of the UTRAN 3 is given in Technical Specification TS 25.401 V3.2.0 of the 3rd Generation Partnership Project. Figure 1 also illustrates a mobile terminal or User Equipment (UE) 6, a Serving GPRS Support Node (SGSN) 7 and a GPRS Gateway Support Node (GGSN) 8. The SGSN 7 and the GGSN 8 provide packet switched data services to the UE 6 via the UTRAN (with the GGSN being coupled to the Internet 9).

User data received at an RNC from the UTRAN core network is stored at a Radio Link Control (RLC) entity in one or more RLC buffers. Figure 2 illustrates certain radio protocol layers implemented at the RNC (and the UEs). User data generated at a UE is stored in RLC buffers of a peer RLC entity at the UE. User data (extracted from the RLC buffers) and signalling data is carried between an RNC and a UE using Radio Access Bearers (RABs). Typically, a UE is allocated one or more Radio Access Bearers (RABs) each of which is capable of carrying a flow of user or signalling data. RABs are mapped onto respective logical channels. At the Media Access Control (MAC) layer, a set of logical channels is mapped in turn onto a transport channel. Several transport channels are in turn mapped at the physical layer onto one or more physical channels for transmission over the air interface between a Node B and a UE.

It is envisaged that UMTS networks will be widely used for carrying data traffic (and using the services of a SGSN and a GGSN). Most current applications which make use of packet switched data services use the Transport Control Protocol (TCP) in conjunction with Internet Protocol (IP) - TCP is used to provide a (connection-oriented) reliable service over the unreliable IP service. It can therefore be expected that the majority of data communications across a UMTS network will use TCP/IP. The same is true for other mobile communication networks (e.g. GSM, GSM with GPRS

enhancement, EDGE), although this discussion is restricted to UMTS merely to illustrate the applicability of the present invention.

Considering the transfer of data in the downlink direction (i.e. from the UTRAN to the UE), signalling and user data packets (e.g. IP packets) destined for the UE are passed, via a PDCP entity, to the Radio Link Control (RLC) entity. The RLC is responsible for the segmentation of packets (as well as for certain error correction and ciphering functions), and generates RLC Protocol Data Units (PDUs) which are passed to the MAC layer and received as MAC Service Data Units (SDUs). The MAC layer schedules the packets for transmission.

Where a UE has been allocated a dedicated channel (DCH) or downlink shared channel (DSCH), the MAC-d PDUs are passed to the Node B for transmission over the air interface. However, where the UE has been allocated a common channel, the MAC-d PDUs are passed to a MAC-c entity and are received thereby as MAC-c SDUs. The MAC-c entity schedules MAC-c PDUs for transmission on the common channel.

It is assumed now, by way of example, that the UE 6 has requested the downloading of IP data from a correspondent node (CN) 10 which is coupled to the Internet 9. The request is sent via the UMTS network 1 and the Internet 9. The request may be initiated for example by the user of the UE 6 entering a URL into a web browser application at the UE 6. Upon receiving the request, the CN 10 identifies the necessary data, and the TCP entity at the CN 10 begins transmitting IP data packets to the UE 6 using the slow start mode described above. Assuming that there is no congestion in the transmission link, the sending rate will increase until the congestion avoidance mode is entered (the rate may increase further thereafter).

IP data packets are routed through the Internet 9, the core network 2, and the UTRAN 3 to the RNC 4 serving the UE 6. IP packets arriving at the RLC layer are placed in an allocated RLC buffer, awaiting transmission to the UE 6 over the radio interface using a common channel or, more preferably, a dedicated channel. It is noted that several TCP/IP connections may be simultaneously active over one allocated logical channel for a given UE, in which case all IP packets associated with these connections and

travelling in the downlink direction will be placed in the same RLC buffer. Alternatively, different connections may be mapped to different logical channels in which case the UE is associated with several RLC buffers simultaneously. A TCP connection may have some guaranteed quality of service or may rely on so-called "best effort". The following discussion concerns best effort connections.

As explained above, a sudden burst of IP packets from a TCP sender (i.e. at the CN 10) may cause the radio link between the RNC 4 and the UE 6 to become congested. There is then a danger that the RLC buffer will become full resulting in the dropping of packets which would in turn result in the TCP sender remaining in or dropping back into the slow start mode. It is desirable to avoid this happening as it results in perceived poor performance on the part of the user, and an inefficient use of the link bandwidth.

In order to avoid this problem, and in particular to provide early notification to the TCP sender of congestion in the link, the algorithm illustrated in the flow diagram of Figure 3 is used to control the RLC buffer queue. The algorithm uses three queue threshold levels, a fixed minimum threshold level T_{min}, a fixed maximum threshold level T_{max}, and a movable or variable threshold level T_{drop}. T_{drop} is initially set to T_{min}.

As each packet arrives at the RLC layer of the serving RNC 4, the size q of the RLC buffer queue is determined. If the queue size q is greater than T_{max}, the queue is large relative to the link capacity and it can be assumed that the link is congested. The received packet is therefore dropped. It is then determined whether T_{drop} is less than T_{max}. If so, the value of T_{drop} is incremented by some predetermined hysteresis value Δ. If on the other hand T_{drop} exceeds T_{max}, T_{drop} is not incremented. Assuming that subsequent packets are delivered to the UE 6, the TCP sender will receive duplicate acknowledgements notifying it of the missing packet. The fast retransmit mechanism will be used to resend that packet.

If it is determined that the queue size q is less than T_{max}, but that the queue size q is greater than T_{drop}, the received packet is still discarded and T_{drop} is incremented if T_{drop} is less than T_{max}. However, if the queue size q is less than T_{max}, and less than T_{drop}, it can be assumed that the link is not congested and the received packet is accepted and

placed at the back of the queue in the RLC buffer. If the queue size is then determined to be less than T_{drop} by some predetermined amount, $(1+H_d)\Delta$, and T_{drop} is greater than T_{min}, it can be assumed that an earlier congestion has been eased. T_{drop} is therefore decremented by the hysteresis value Δ. If either of these conditions is not met, T_{drop} is left unchanged.

The value Δ may be the same for both incrementing and decrementing steps. However, some advantage may be obtained by decrementing T_{drop} by a value which is smaller than that used when incrementing T_{drop}. This tends to result in T_{drop} being decremented more frequently, but with a smaller "granularity", than would otherwise be the case.

It will be clear from the preceding discussion that at early detection of congestion, i.e. when the queue level exceeds T_{drop}, one packet is discarded. The queue threshold mark T_{drop} is then increased by the hysteresis value Δ. This value of the moving threshold T_{drop} is valid until the queue is either drained by an amount $H_d \cdot \Delta$ or filled by an amount Δ. In the event that the queue is drained by $H_d \cdot \Delta$, the moving threshold is decreased by Δ. The parameter H_d is used to define an asymmetric switch and should be greater than 0.

It will be noted that there are four parameters which must be set; T_{min}, T_{max}, Δ, and H_d. However, if a symmetric threshold switching is used (i.e. H_d=1), there are only three settable parameters.

Parameter T_{min}: The setting of the early congestion threshold mark T_{min} is a critical issue. This parameter defines the queue capacity that should accumulate both high-frequency variations due to packet bursts and low frequency variations caused by the TCP bandwidth probing mechanism. One possible way of determining T_{min} is as follows:

The link capacity is estimated according to

$$LC = (RTT_{wc} + RTT_{link}) \cdot DR$$

where RTT_{wc} is the worst-case estimate of the TCP roundtrip time without the contribution from the wireless (bottleneck) link. RTT_{link} is the delay contribution from the congested link and *DR* denotes the link data rate.

Example: A reasonable estimate for RTT_{wc} could be 200 - 300ms whereas a wireless link may exhibit some 200-400 ms for RTT_{link}. The total TCP RTT, excluding buffering, is then some 0.4 – 0.7 s. *LC* is the capacity of the link, *excluding* buffering capacity prior to the link. To ensure link utilisation, it should be ensured that the TCP window (load) is greater than (or equal to) *LC*. Excessive load is stored in the queue. Since the TCP window is halved at detection of congestion, it should be ensured that the TCP window may grow to 2*LC*. A queue capacity of *LC* guarantees that the TCP load can vary between *LC* and 2*LC*, provided TCP timeouts can be prevented. A constant ε may added to the congestion threshold mark:

$T_{\min} = LC + \varepsilon$.

The parameter ε should take into account the uncertainty in the estimate of *LC*, as well as the high-frequency variations in the queue level due to packet bursts. The parameter can be set to zero or to a positive value to account for a small number of packets, depending on how conservative the LC estimate is.

The parameter T_{max}: Setting of T_{max} is less critical, as a well behaving queue should not reach this fill state in normal operation. Thus, T_{max} should be large - without wasting hardware resources. A minimum requirement is that the queue should be able to accommodate the load increase during *slow-start* clocked by unacknowledged TCP segments in flight *prior* to the segment that was discarded from the queue at congestion detection. This reasoning would support a minimum value of $T_{max} = 2 \cdot T_{min}$ for a queue in which the arriving packet is subject to discard. However, a value $T_{max} = 4 \cdot T_{min}$ may be used.

The threshold Δ should be set to account for occasional bursts of incoming packets, *(i.e. some 3-5kbytes).*

Figure 4 illustrates an alternative mechanism for controlling the RLC buffer. This mechanism relies on only two fixed thresholds, T_{max} and T_{min}. This is similar to the RED mechanism. However, rather than use a probabilistic approach to dropping packets when the queue size lies between T_{max} and T_{min}, a counter C is used to allow only one in every (n+1)th packet to be dropped. The parameters T_{max} and T_{min} may be determined as described above (with reference to the mechanism of Figure 3). The counter value should be related to the expected TCP window size to avoid discarding several segments from the same TCP window – preferred values are in the range of 20 to 30 packets. The expected number of TCP connections sharing the link bandwidth may, however, affect the preferred settings.

Figure 5 illustrates yet another mechanism for controlling the RLC buffer. This mechanism differs from that illustrated in Figure 4 in so far as, when reducing T_{drop}, T_{drop} is made to track the queue size, always exceeding the queue size by a fixed amount Δ_2. The value Δ_2 may or may not be the same as the value Δ_1 by which T_{drop} is incremented. The advantage of this approach is that if the queue size falls rapidly, T_{drop} will also fall rapidly ensuring that the new value of T_{drop} is appropriate when further packets are subsequently received.

The mechanisms described above have assumed that when a decision is made to drop a packet, the packet which is dropped is that packet most recently received from the TCP sender. However, it may be advantageous to accept this packet, adding it to the back of the queue, whilst dropping a packet already in the queue, preferably at or close to the front of the queue. Dropping packets in this way will most probably speed up the notification of congestion to the sender – subsequent packets of the same TCP connection may already be in the queue behind the dropped packets resulting in the rapid return of duplicate acknowledgements to the sender. This approach also reduces the chances of the dropped packet being the last packet in a transmission, for which no duplicate acknowledgements can be returned (and to which fast retransmission cannot be applied).

It will be appreciated by the person of skill in the art that various modifications may be made to the above described embodiments without departing from the scope of the

present invention. In particular, whilst the above embodiments have been concerned with the transfer of data in the downlink direction, the invention applies equally to the transfer of data in the uplink direction, i.e. from the UE to a correspondent node. In this case, the RLC buffer being controlled will be a buffer associated with the radio layers at the UE. It will also be appreciated that the invention is not limited to applications in UMTS networks but also finds applications in other packet networks where data is buffered including, but not limited to, other telecommunications networks.

Claims

1. A method of controlling the entry of data packets into a buffer present in a packet transmission link, the method comprising:

defining a first fixed threshold level and a second variable threshold level for the packet queue size within the buffer; and

for each data packet arriving at the buffer, performing a congestion avoidance procedure if the current buffer queue size exceeds said first or second threshold level, and adjusting said second variable threshold level depending upon (a) whether or not a packet is dropped and (b) upon the relative values of the first and second thresholds and the queue size.

2. A method according to claim 1 and comprising initialising the second variable threshold level to a predetermined minimum threshold level which is less than said first fixed threshold level.

3. A method according to claim 1 or 2, wherein the second variable threshold level is adjusted by incrementing or decrementing the level by a fixed amount.

4. A method according to any one of the preceding claims, wherein the amount by which the variable threshold is incremented is the same as the amount by which it is decremented.

5. A method according to any one of claims 1 to 3, wherein the amount by which the variable threshold is incremented is greater than the amount by which it is decremented.

6. A method according to any one of claims 1 to 3, wherein, when the variable threshold is incremented it is incremented by a fixed amount and, when the variable threshold is decremented, it is decremented to within some predetermined value in excess of the queue size so as to track the queue size.

7. A method according to any one of the preceding claims, wherein the second variable threshold level is incremented following receipt of a packet if said congestion avoidance procedure is performed and the second variable threshold level does not exceed the first threshold level.

8. A method according to claim 7, wherein, if said congestion avoidance procedure is performed and the second variable threshold level does exceed the first threshold level, the second variable threshold level is not changed.

9. A method according to any one of the preceding claims and comprising decrementing the second variable threshold level following receipt of a packet if said congestion avoidance procedure is not performed, the queue size is less than the second variable threshold level by some predefined amount, and the second variable threshold level is greater than said minimum threshold level.

10. A method according to claim 9, wherein, if said congestion avoidance procedure is not performed and the queue size exceeds the second variable threshold less said predefined amount, or the second variable threshold level is less than said minimum threshold level, the second variable threshold level is not changed.

11. A method according to any one of the preceding claims, wherein said congestion avoidance procedure comprises dropping the newly arrived packet or a packet already held in the buffer.

12. A method according to any one of claims 1 to 10, wherein said congestion avoidance procedure comprises marking including in the packet a congestion marker.

13. A method according to any one of the preceding claims, wherein the IP packets belong to a TCP/IP connection, with the packets arriving at the buffer being transmitted there by a TCP sender.

14. A method according to any one of the preceding claims, wherein the buffer is be associated with a wireless communication network

15. Apparatus for buffering data packets in a packet transmission link, the apparatus comprising:

a buffer for containing a queue of data packets;

an input for receiving data packets;

a memory storing a first fixed threshold level and a second variable threshold level for the packet queue within the buffer; and

a controller arranged, for each data packet arriving at the buffer, to perform a congestion avoidance procedure if the current buffer queue exceeds said first or second threshold level, and to adjust said second variable threshold level depending upon (a) whether or not a packet is dropped, and (b) the relative values of the first and second thresholds and the queue size.

16. A method of controlling the entry of data packets into a buffer associated with the radio layers of a wireless network, the buffer storing packets prior to their transmission over the radio interface, the method comprising:

defining minimum and maximum threshold levels for the packet queue within the buffer; and

for each data packet arriving at the buffer,

performing a congestion avoidance procedure if the current buffer queue exceeds said maximum threshold level; or

not performing said procedure if the current buffer queue is less than said minimum threshold level; or

if the current buffer queue lies between said maximum and minimum thresholds, performing said procedure only if the arriving packet is the *n*th packet arriving at the buffer since the arrival of the last packet which caused said procedure to be performed.

17. Apparatus for use in a wireless network and comprising:

a buffer for storing data packets

an input for receiving data packets;

a memory storing minimum and maximum threshold levels for the packet queue within the buffer; and

a controller arranged for each data packet arriving at the buffer to,

perform a congestion avoidance procedure if the current buffer queue exceeds said maximum threshold level; or

not perform said procedure if the current buffer queue is less than said minimum threshold level; or

if the current buffer queue lies between said maximum and minimum thresholds, performing said procedure only if the arriving packet is the *n*th packet arriving at the buffer since the arrival of the last packet which caused said procedure to be performed.

Amendments to the claims have been filed as follows

1. A method of controlling the entry of data packets into a buffer present in a packet transmission link, the method comprising:

defining a first fixed threshold level and a second variable threshold level for the packet queue size within the buffer; and

for each data packet arriving at the buffer, performing a congestion avoidance procedure if the current buffer queue size exceeds said first or second threshold level, and adjusting said second variable threshold level depending upon (a) whether or not a packet is dropped and (b) upon the relative values of the first and second thresholds and the queue size.

2. A method according to claim 1 and comprising initialising the second variable threshold level to a predetermined minimum threshold level which is less than said first fixed threshold level.

3. A method according to claim 1 or 2, wherein the second variable threshold level is adjusted by incrementing or decrementing the level by a fixed amount.

4. A method according to any one of the preceding claims, wherein the amount by which the variable threshold is incremented is the same as the amount by which it is decremented.

5. A method according to any one of claims 1 to 3, wherein the amount by which the variable threshold is incremented is greater than the amount by which it is decremented.

6. A method according to any one of claims 1 to 3, wherein, when the variable threshold is incremented it is incremented by a fixed amount and, when the variable threshold is decremented, it is decremented to within some predetermined value in excess of the queue size so as to track the queue size.

7. A method according to any one of the preceding claims, wherein the second variable threshold level is incremented following receipt of a packet if said congestion avoidance procedure is performed and the second variable threshold level does not exceed the first threshold level.

8. A method according to claim 7, wherein, if said congestion avoidance procedure is performed and the second variable threshold level does exceed the first threshold level, the second variable threshold level is not changed.

9. A method according to claim 2 or to any one of claims 3 to 8 when appended to claim 2 and comprising decrementing the second variable threshold level following receipt of a packet if said congestion avoidance procedure is not performed, the queue size is less than the second variable threshold level by some predefined amount, and the second variable threshold level is greater than said predefined minimum threshold level.

10. A method according to claim 9, wherein, if said congestion avoidance procedure is not performed and the queue size exceeds the second variable threshold less said predefined amount, or the second variable threshold level is less than said predefined minimum threshold level, the second variable threshold level is not changed.

11. A method according to any one of the preceding claims, wherein said congestion avoidance procedure comprises dropping the newly arrived packet or a packet already held in the buffer.

12. A method according to any one of claims 1 to 10, wherein said congestion avoidance procedure comprises marking including in the packet a congestion marker.

13. A method according to any one of the preceding claims, wherein the IP packets belong to a TCP/IP connection, with the packets arriving at the buffer being transmitted there by a TCP sender.

14. A method according to any one of the preceding claims, wherein the buffer is be associated with a wireless communication network

15. Apparatus for buffering data packets in a packet transmission link, the apparatus comprising:

a buffer for containing a queue of data packets;

an input for receiving data packets;

a memory storing a first fixed threshold level and a second variable threshold level for the packet queue within the buffer; and

a controller arranged, for each data packet arriving at the buffer, to perform a congestion avoidance procedure if the current buffer queue exceeds said first or second threshold level, and to adjust said second variable threshold level depending upon (a) whether or not a packet is dropped, and (b) the relative values of the first and second thresholds and the queue size.

Application No: GB 0113214.1
Claims searched: 1 to 15
Examiner: Daniel Voisey
Date of search: 19 December 2001

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H4K (KTKX)

Int Cl (Ed.7): H04L 12/24, 12/56, 29/06

Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	EP 1028600 A2	(NEC) see page 2 line 44 to page 3 line 15.	
A	US 6134239	(HEINANEN) see column 3 line 64 to column 4 line 40.	
A	US 6034945	(HUGHES) see column 2 lines 14 to 38.	
A	US 5546389	(WIPPENBECK) see column 2 lines 26 to 40.	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.